

## CephFS - Feature #44279

client: provide asok commands to getattr an inode with desired caps

02/24/2020 09:37 PM - Patrick Donnelly

<b>Status:</b>	Fix Under Review	<b>% Done:</b>	0%
<b>Priority:</b>	Normal		
<b>Assignee:</b>	Kotresh Hiremath Ravishankar		
<b>Category:</b>			
<b>Target version:</b>	v17.0.0		
<b>Source:</b>	Development	<b>Affected Versions:</b>	
<b>Tags:</b>		<b>Component(FS):</b>	Client, qa-suite
<b>Backport:</b>		<b>Labels (FS):</b>	qa
<b>Reviewed:</b>		<b>Pull request ID:</b>	35992
<b>Description</b>			
Idea is to avoid using UNIX commands or libcephfs calls to implicitly acquire the caps we want. Instead, write a command that explicitly tries to acquire caps for an inode.			
This is a first step to creating a test suite for cap acquisition/revoke behavior with the MDS.			

### History

#### #1 - 03/02/2020 02:45 PM - Greg Farnum

We may also just want to bring the SynClient or do something directly with libcephfs to grab exact sequences instead of going through the ceph-fuse asok.

#### #2 - 06/08/2020 02:22 PM - Patrick Donnelly

- Assignee set to Kotresh Hiremath Ravishankar

#### #3 - 06/30/2020 01:40 PM - Jeff Layton

Do we need an asok interface for this? If you're planning to write testcases that link in libcephfs directly, then you can just add new libcephfs calls that do what you want, and write testcases that just call them.

The main reason to add something like asok interface would be if you want to be able to issue these sorts of requests in conjunction with (e.g.) a FUSE mount or ganesha export. Are you planning to do that?

#### #4 - 06/30/2020 06:14 PM - Patrick Donnelly

Jeff Layton wrote:

Do we need an asok interface for this? If you're planning to write testcases that link in libcephfs directly, then you can just add new libcephfs calls that do what you want, and write testcases that just call them.

The main reason to add something like asok interface would be if you want to be able to issue these sorts of requests in conjunction with (e.g.) a FUSE mount or ganesha export. Are you planning to do that?

The eventual goal is to test the expected behavior of two clients issuing getattr requests. Yes, we could simulate with UNIX syscalls like `mkdir` but that relies on second order effects -- what caps the `mkdir` operation will try to fetch (which may touch many inodes in the path lookup). It's better to directly test the behavior you're trying to confirm without any extraneous operations.

**#5 - 07/09/2020 07:22 AM - Kotresh Hiremath Ravishankar**

- Status changed from New to In Progress

**#6 - 07/09/2020 12:09 PM - Kotresh Hiremath Ravishankar**

- Pull request ID set to 35992

**#7 - 07/09/2020 04:56 PM - Patrick Donnelly**

- Status changed from In Progress to Fix Under Review

**#8 - 03/17/2021 09:46 PM - Patrick Donnelly**

- Target version changed from v16.0.0 to v17.0.0